

COP

IN THE CLAIMS:

Please cancel claim 2-3, 7, and 14; add claim 15; and amend claims 1, 4-6, 8, 10 and 13. All claims are reproduced below.

1 1. (Currently amended) A computer-implemented method for efficiently
2 parsing input data, comprising:
3 receiving a data file;
4 retrieving a stored version of the data file and a template/token tree
5 corresponding to the data file, the tree including at least one static node;
6 comparing the stored version of the data file with the received data file to
7 identify non-matching content in the received data file;
8 parsing only the non-matching content of the received data file to form at least
9 one subtrees subtree;
10 replacing at least one static node of the template/token tree with a token; and
11 creating a mapping from each token to one of the template/token tree to the
12 subtrees.

1 2. (Canceled)

1 3. (Currently amended) The computer-implemented method of claim 1 wherein
2 creating the mapping from the tree to the subtrees further comprises:
3 adding at least one token node to the template/token tree; and
4 creating a mapping from each token to at least one subtree.

1 4. (Currently amended) The computer-implemented method of claim 1 wherein
2 the data file is a web page.

COPY

1 5. (Currently amended) The computer-implemented method of claim 1 wherein
2 the data file is an HTML file.

1 6. (Currently amended) A method for efficiently parsing web pages,
2 comprising:
3 receiving a first HTML page;
4 retrieving a cached version of the HTML page and a template/token tree
5 corresponding to the first HTML page, the tree including at least one static
6 node;
7 comparing the cached version of the HTML page with the received HTML page
8 to identify non-matching content in the received HTML page;
9 parsing only the non-matching content in the received HTML page to form at
10 least one subtree;
11 replacing at least one static node of the template/token tree with a token; and
12 creating a mapping from ~~the template/token tree to each token to one of the~~
13 subtrees.

1 7. (Canceled)

1 8. (Currently amended) A method for efficiently parsing HTML pages,
2 comprising:
3 receiving a first HTML page;
4 responsive to a determination that a cached version of the HTML page exists:
5 retrieving the cached version of the HTML page and a first
6 template/token tree corresponding to the first HTML page, the
7 first tree including at least one static node;

COPY

8 comparing the cached version of the first HTML page with the
9 received HTML page to identify non-matching content in the
10 received HTML page;
11 parsing only the non-matching content to form a subtree;
12 creating a mapping from a token of the first tree to associating the
13 first tree and the subtree;
14 responsive to a determination that the cached version of the HTML page does
15 not exist;
16 parsing the received HTML page to form a second template/token
17 tree, the second tree containing at least one static node; and
18 storing the second tree and the received HTML page.

1 9. (Original) A method for providing derivative services comprising:
2 receiving a first HTML page;
3 constructing a template/token tree from the received HTML page, the tree
4 comprising a plurality of nodes;
5 determining that at least one node of the tree contains static content;
6 determining that at least one node of the tree contains dynamic content;
7 replacing the nodes of the tree containing dynamic content with tokens;
8 parsing the dynamic content to form subtrees; and
9 mapping the tokens to the subtrees.

1 10. (Currently amended) A computer-implemented method of providing
2 derivative services, comprising:
3 receiving a request for derivative services content from a customer;
4 retrieving data from a plurality of primary service providers on behalf of the
5 customer, by:

22271/05227/5F/5036911.9

COPY

6 identifying static content that has been previously retrieved from the
7 primary service providers and stored, and corresponding
8 template/token trees that have also been stored;
9 identifying dynamic content that differs from the previously retrieved
10 content;
11 parsing the dynamic content to form subtrees;
12 adding tokens to the template/token trees;
13 mapping the tokens to the subtrees;
14 creating at least one content page comprising the retrieved data; and
15 providing the created pages to the customer.

1 11. (Original) A method for efficiently parsing input data, comprising:
2 receiving a first data file;
3 retrieving a stored template/token tree, the stored template/token tree having
4 content associated with the first data file and containing at least one static
5 node and at least one token;
6 retrieving a second data file, the second data file associated with the first data
7 file;
8 identifying non-matching content present only in the first data file;
9 parsing only the non-matching content of the first data file to form at least one
10 subtree; and
11 mapping at least one of the tokens to at least one of the subtrees.

1 12. (Original) The method of claim 11, further comprising:
2 responsive to identifying non-matching content present only in the first file:
3 adding at least one new token to the template/token tree.

22271/05227/SF/5036911.9

COPY

1 13. (Currently amended) A system for efficiently parsing input data,
2 comprising:
3 at least one virtual browser for retrieving content from primary content servers;
4 an identification engine, communicatively coupled to the virtual browser for
5 identifying retrieved content;
6 a cache, communicatively coupled to the virtual browser and the parsing engine,
7 for storing retrieved content and template/token trees;
8 a comparison engine, coupled to the virtual browser for comparing retrieved
9 content with stored content to identify differing content not stored in the
10 cache;
11 a token master, communicatively coupled to the cache, for allocating new tokens
12 to the virtual browser;
13 a parsing engine, communicatively coupled to the virtual browser, for parsing
14 content identified by the comparison engine as differing content and forming
15 subtrees from the content and creating a mapping from new tokens to
16 formed subtrees; and
17 a content server, coupled to the virtual browser.

1 14. (Canceled)

1 15. (New) A computer program product for efficiently parsing input data, the
2 computer program product stored on a computer-readable medium and including
3 instructions for causing a computer to carry out the steps of:
4 receiving a data file;
5 retrieving a stored version of the data file and a template/token tree
6 corresponding to the data file, the tree including at least one static node;

COPY

- 7 comparing the stored version of the data file with the received data file to
- 8 identify non-matching content in the received data file;
- 9 parsing only the non-matching content of the received data file to form at least
- 10 one subtree;
- 11 replacing at least one static node of the template/token tree with a token; and
- 12 creating a mapping from each token to one of the subtrees.